White Paper: SCORM 2.0 LETSI Call for Participation

SCORM2: redefining reusable educational activities

Rachel H. Ellaway PhD

Introduction

SCORM is summed up in the first two words of its acronym; shareable content. This is both its strength and its weakness. A strength in that it has been the most successful and widely implemented interoperability specification in education (except perhaps RSS) therefore clearly addressing the 'sharing' goal. A weakness in that the assumptions and perhaps limits of SCORM1 and the philosophy behind it equated education with access to and interaction with content, moreover content typically expressed as static webpages and media objects that can be embedded within these webpages. This perspective has its roots in two traditions; firstly tutors primarily expressing their courses as written materials, notes and guides, and secondly a cognitive focus on knowledge acquisition and transfer. It also comes from a preemptive focus on one kind of technology – content primarily expressed as text or providing the con-text for everything else around it. In the meantime the educational impact of new technologies has found much more focus and uptake in social, collaborative and activity-focused forms although many still have a content focus, particularly Web 2.0 services like Wikipedia, YouTube and Flickr.

We now know more, can do more and can posit a richer, more adaptable and far more learner- and cognitive-focused (rather than teacher- and authority-focused) model for sharing materials between systems. This white paper presents a vision of the directions that SCORM will take in its next iteration to transcend content as knowledge to embrace a common model of content as activity that spans both the acquisition and application of knowledge.

The proposed model for SCORM2 is a three stream model dynamically combining expressions of the content used, the context in which it is presented to the user and the controls (rules etc) layered in to make the first two come alive in an educational setting.

Background

All technologies embody preemptive design choices that shape the realities of the environment in which they function. The most fundamental construct of SCORM is arguably the package. It defines and constrains what SCORM can and cannot be and do. The direct impact on its use on education has been both to enable exchange and reuse of static content and at the same time to establish a very basic model as the dominant design for educational materials, not least because of its support and high visibility from commercial and government implementations. In the meantime content-based e-learning has been outstripped in many quarters by serious games, collaborative tools (wikis, blogs, discussion) and in particular by third-party Web 2.0 content services such as YouTube, Flickr, Facebook, and iTunes.

There are two levels involved or at least implied by all of these approaches: interoperability is the ability for two or more systems to exchange data in non-coupled ways that preserve its syntactic and semantic qualities. Integration on the other hand is the ability for two or more system to exchange data in coupled ways (typically synchronously). Both of these depend on data standards, which in turn involve many participants from many different contexts engaged on the collaborative process of combining, abstracting and weaving their different needs together (Ellaway, 2006).

From a SCORM perspective the focus has been on interoperability with no explicit runtime dependence between the system that created a package and the one that runs it. In the intervening years since SCORM was first designed a number of significant technical

developments have at least potentially redefined its environment including; web services, APIs and AJAX, and non-web platforms such as Adobe's Air and Flash. So we have SCORM1 eclipsed in terms of educational utility by other technologies and SCORM's technical environment also changed. The implications would seem to be that package-based educational models are now redundant.

However, that is not the position taken in this white paper. There are already many interoperability and integration models in place or under development across the educational technology spectrum (see for instance www.e-framework.org) but the ability to package all of the instructions and components of an educational activity and have it run in a compliant runtime environment is still much needed, the changes we propose are in the nature of the contents of a package and how it can be made significantly more educationally relevant and adaptable than the current model.

Educational Affordances

Let's review the breadth of educational models and objectives SCORM might cover. Bloom's model of educational objectives (1984) describes a continuum from the acquisition and manipulation of knowledge through the application of that knowledge to the evaluation and creation of new knowledge. Similarly Miller in describing clinical competence (1990) describes a pyramid building from 'knows' through 'knows how' and 'shows how' to 'does' see figure 1.

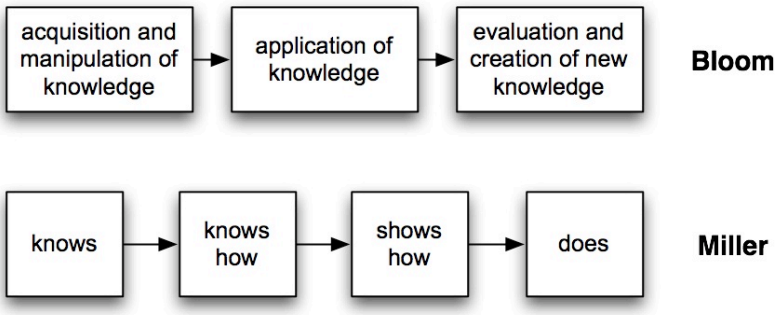


Figure 1: the Bloom and Miller models of learning outcomes/objectives

At present SCORM, by concentrating on sequenced webpages and a content/knowledge model, typically only supports designs that afford outcomes at the base/left of these educational continua. Going beyond knowledge transfer involves developing a model that can be used across the cognitive spectrum.

The application of knowledge is well described in a number of educational models including the 'practicum' (Schön, 1987), game-informed learning (Begg, Dewhurst et al. 2005) and problem-based learning. All of these involve some form of simulation which in turn should provides a safe environment, that is forgiving of mistakes, controlled, structured, proactive, reproducible, standardized, and objective and most important is able to provide structured, meaningful and rich feedback (Issenberg, 2006).

Although educational modeling languages, and in particular IMS' Learning Design, have previously tried to provide a common framework for different educational designs they haven't focused down to the runtime model and as a result their uptake has been fairly limited. To be successful SCORM 2 needs to have enough adaptability and abstraction to encompass all sorts of different activities that cover as much of these outcome continua as possible.

The consideration of the primacy of feedback raises one further consideration before moving on. SCORM is about encoding and transport of educational materials not about authoring them.

Educational concerns such as the quality or quantity of feedback or other issues such as cognitive load are therefore out of scope. SCORM 2 must be able to afford the selection and use of different strategies but as far as possible not preempt or constrain them.

Building on SCORM1

Phil Dodds, SCORM's original architect and champion, described SCORM1 as having a number of key characteristics (http://adlcommunity.net/mod/resource/view.php?id=458) :

- SCORM1 standardizes how to launch and track directed learning experiences, and to define the intended behavior and logic of complex learning experiences so content can be reused, moved, searched for, and recontextualized.
- SCORM1 has three parts: 1. Overview - about the model, vision and future, 2. Content Aggregation Model - how to put learning content together so it can be moved and reused, and 3. Run Time Environment: how content is launched and the learner's progress is tracked and reported back.
- The "organization" part of packaging is the blueprint for the design of a particular learning experience. It tells an LMS what the designer intended when the learning experience was authored.
- The data model standardizes how LMS systems track learners.
- Sequencing is crucial to representing complex behaviors of learning experiences in a standardized way.

Pretty much all of this 20,000 foot view remains relevant to SCORM2. To accommodate legacy burdens SCORM2 without compromising its new strengths should be able to accommodate SCORM1 designs, at least in terms of what they can do rather then how they do it. SCORM2 needs to be: as conceptually simple as SCORM1 or more so, more adaptable and education/cognitively focused, more accommodating of developing online educational theory and practice, practical, accessible, implementable and technically expressable, as far as possible based on common standards and interoperability frameworks.

Proof of Concept

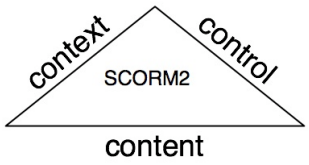
There is nothing blue skies or speculative about the proposed design for SCORM2. Much of the thinking set out in this white paper arose from the development of the OpenLabyrinth platform (download from http://sourceforge.net/projects/openlabyrinth). This tool, originally developed to support authoring, playback and analysis of educational virtual patient narratives uses nodes and links, counters and media object references throughout.

The same thinking underpins MedBiquitous' Virtual Patient (MVP) data specification (currently going through ANSI accreditation) – see http://medbiq.org/working_groups/virtual_patient - which is being adopted by most of the commonly used virtual patient tools in academic use. It is even being used in Second Life – see http://www.youtube.com/watch?v=6O-vzhy-oJ8 for the excellent work being done by Daden in the UK - the key additional requirement being able to target interactions to specific objects in a 3D world i.e. an html link is replaced with a Second Life artifact that triggers the next step in activity.

SCORM 2 Architecture

Drawing on these two existing models the proposed architecture for SCORM 2 is based on three interconnected strands: context, content and control – these are designed to afford the greatest balance between simplicity, adaptability and comprehensiveness with regards to at least

minimally content-based with predefined properties and ways of executing. Compare this for instance with forums, wikis and blogs, which are by definition emergent with no predefined content or structure beyond the supporting systems' affordances.



CONTEXT

Context concerns the structure of the activity, materials or any other artifact represented by the SCORM 2 package. The simplest and most abstract way of representing relationships between discrete parts of a whole is to use a nodal architecture comprising one or more nodes and the links between them. This is for instance the model used in SGML/ XML, networks, topic maps, taxonomies, HyperCard, the Web, curricula, wikis and across many branches of academe.

There would be two levels of nodal architecture:

- *Presentation node model:* this is about how the user is presented with material based on a mixture of author definitions and a user's previous actions. It represents the paths that a user can take but not how – this is addressed in CONTROL. In this model the direction of the links (from A to B but not necessarily from B to A) is crucial. In this respect it is based on the mathematical principles of acyclic graphs. It also has the advantage of being familiar and easy for developers, implementers and educationalist to comprehend. Each node in the *presentation node model* is equivalent to a screen, page or other unit of display and is therefore best considered as a container for content. The logical links between nodes are expressed as pairs of start and end node IDs along with any additional properties they might have. Note that users may be redirected from one node to another without there being an overt link expressed by dint of a CONTROL rule being executed. Links between presentation nodes, while not therefore being essential are the norm. Note that an activity could be assembled using only non-connected nodes and CONTROL rules (see use cases).
- *Organization node model:* this is about how parts of the *presentation node model* are grouped, menus are specified and other aspects of the meta-organization of what the user can do are set out. Each node in the *organization node model* is a container for one or more presentation nodes along with their properties. Links between organization nodes are likely to be less commonly used.

Between these two nodal models CONTEXT supersedes the simple sequencing model used for SCORM 1 by combining sequencing and organization into a more flexible and abstract model for both.

CONTENT

Content is (not surprisingly) concerned with the materials presented to the user or with which the user interacts. There are four types of content objects:

1. *Text:* any pure text elements, which can therefore be directly expressed within XML containers. There are several candidate formats including HTML, XHTML and the simplest (although non-standardized and therefore as yet untapped potential) of wiki formatting. Particular strengths of a standardized wiki format would be its non-clash with XML characters (as opposed to the <, > and & in HTML and XHTML) and the ability to dynamically embed all sorts of media objects with simple object references.

2. *File objects:* typically binary objects (and therefore non-XML expressable/parsable) including images, movies, audio files and binary document files such as DOC, PPT and XLS. The defining aspect of file objects is that they are non-interactive i.e. they do not change dynamically in response to user input, they do not connect to an external data source etc – see dynamic objects. Note that presentation files such as CSS and JavaScript would also be tracked here. Additional parameters are limited to presentation such as width and height for images and movies
3. *Dynamic objects*: these are any media object that changes itself or its environment depending on a user's actions. Examples could include Flash files, Java Applets, MathML readers, AJAX services, webservices and dependencies on messaging with learning management and other enterprise systems. Additional parameters would include both presentational and runtime elements such as data sources, service details etc.
4. *Content aggregates*: these are any (presentational) construct of the first three types. For instance an aggregate may combine text, images and a search service that can be reused as a single aggregate in several places in the presentation node model. Aggregates may or may not include presentation syntax (e.g. a table setting out which element is presented where).

Content is intrinsically part of SCORM1 (it's in the acronym stupid) with its package manifest and profound ties to HTML. This proposed new CONTENT model extends the basis for using content by making it more addressable and reusable than before.

CONTROL

Control is the new component with respect to SCORM1 although it incorporates some aspects of the SCORM1 tracking model. Control in educational activities involves the rules, triggers and behaviors that transform static content into educational activities such as lessons, tests, games and simulations. It is this that was missing in SCORM1 (and most other educational technology models) without which the learner is dependent on dumb 'train tracks' through materials, enlivened by authors where they can. Control in SCORM2 consists of two kinds of objects: execution and wrapper:

- *Execution Controls*: these are the objects that dynamically manage the user experience, particularly in response to user input and decisions. Execution objects may be created as global to an activity or local to a node or link and they may be changed as a result of the links clicked or the nodes visited. For instance a counter may be set up global to the activity and its value changed depending on what the user does, the value of that counter may then trigger predetermined rules which in turn change what the user can see and do. The execution objects envisaged include:
    - *Counter Controls*: these are one or more numeric runtime variables that can have their values altered based on the user's actions. For instance a counter for time could be changed based on how long it would take to implement the choices the learner makes. Ranging anywhere between zero and many, a series of counters can be used to track factors such as: whether a decision has a positive or negative effect (or both using different counters) in terms of time, money, credibility etc, picking up and then using objects (keys, maps, health, morale etc), answers right/wrong, decisions right/wrong and number of attempts made. Any one counter may have zero to many counter rules – these are triggered once any given counter has reached a preset value at which the user is redirected (for instance to an end page if the counter drops below zero or to the 'next level' if it gets above an arbitrary success point). Any given counter rule will be triggered globally e.g. evaluate all counters as each presentation node loads unless disabled at a particular node - this is a node property.

- *Conditional Controls*: these are attached to particular presentation nodes and involve parsing which presentation nodes have been visited before the current one in the current activity session. If the required nodes have been visited then the user proceeds, if not then they are given a different message from the default text or they are redirected to a different node. For instance a user might need to have previously requested some information in order to see it later on.
- *Content Controls*: these allow users to interact more directly with content (SCORM2 runtime as opposed to internal interactions such as with Flash objects). Content objects perform two functions: the first object is the trigger object and the second object is the one that changes as a result of the trigger. There are two basic properties:
  - Trigger – this makes a content object interactive so that a user's click (content trigger) can be recorded – binary on/off
  - Display – this is a state marker regarding the way content is displayed: show (default), show as triggered, show after triggered

  As an illustration a node may display a series of options such as questions the user might ask – the text of the question is the trigger object and the answer is displayed at the next node (show after triggered). Only answers for questions selected are shown.

- *Wrapper Controls*: these are services that surround the execution of a learning activity and allow it to report to the user and to any technical systems in which the activity sits or is connected to. There are three kinds of wrapper controls:
  - *Tracking*: this is a record of which nodes have been visited, which content objects have been triggered and the values of all the counters in the current activity session and the time since the activity started that these events took place. This is used by conditional rules and content display rules as well as for providing feedback
  - *Feedback*: this is information given to users to inform them how they've done in the activity – the educational importance of feedback is paramount. Formative feedback (during the activity) is provided in nodes as part of the execution of the activity. The Feedback wrapper control provides summative (at end of activity) feedback based on a number of author configurable feedback rule types:
    - Feedback based on having visited a particular presentation node – typically for 'must visit' or 'must avoid' nodes.
    - Feedback on the time taken to complete the activity as a whole and the time spent on any given presentation node – for the latter too little time would indicate 'gaming' while spending overlong on a critical decision may also be problematic.
    - Feedback on values of any and all of the counters at the end of the session. Many feedback rules may be attached to any given counter.
  - *Reporting*: involves messaging with an external system, this was a basic part of SCORM1. For SCORM2 there are two proposed forms:
    - *Summative Reporting:* this is a report of all the tracked information passed to the external system at the end of an activity.
    - *Formative Reporting:* this is a blow-by-blow report of the current presentation node selected along with all counter and related state information at the time.

Reporting raises a critical conceptual issue with SCORM2 – its expression is intrinsically syntactic; it is up to the teacher (or other players in the context of use) to attach meaning and significance to the reporting/tracking data. This separation ensures SCORM's simplicity and its adaptability to different circumstances. Abstraction and simplicity are essential values in constructing any kind of viable and sustainable systems model and it keeps the intelligence of the network at the edges where it belongs.

Assembling the three SCORM2 strands allows a very wide range of educational activities to be expressed and therefore packaged and exchanged between compliant systems. This abstraction and functional streaming will be essential in maximizing its utility and plurality of application.

SCORM2 Packaging and Runtime

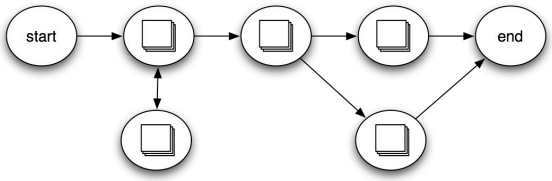
The IMS Content Package (IMSCP) was the basis for SCORM1 packaging itself based around a zip archive containing a manifest of its contents, a folder of media objects, a number of XML files setting out the sequencing, organization and metadata and the XSD files required for validating the XML files. It is proposed that SCORM2 will also use an IMSCP format with an XML files (and XSD) for each of the CONTEXT, CONTENT (manifest) and CONTROL factors.

SCORM1 runtime had several different levels and forms that proved to be confusing required systems to be able to unpack a SCORM package and initiate the starting point. The proposed model for SCORM2 also has runtime components: loading nodes, loading content into nodes and providing links and other information specified in the package, presenting them logically to the user and responding and tracking users' actions and choices. Different implementers may wish to layer in additional local features and services such as forums, wikis or portfolios but the basic runtime requirement, like SCORM1, must be met to be compliant and able to reuse the package.

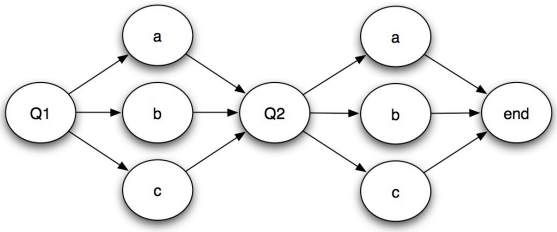
Use Cases

The following brief and highly simplified use cases set out some of the diverse learning activities that can be modeled using this SCORM2 structure (see following diagrams for sample layouts):

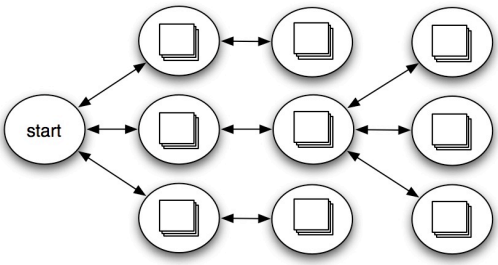
1. Lesson (SCORM1 functionality): a structure of presentation nodes is created representing the required browse paths, content objects are added to each node and the whole lot packaged. When executed this will essentially behave like a SCORM1 package with a start and one or more endpoints.
2. Assessment, quiz, test: questions are set up as branching nodes coming back together for the next question, a counter is created per question and the different nodes allocated different value changes depending on how right or wrong they are. Organization nodes may be used to group presentation nodes per question (stem and choices).
3. Knowledgebase (wiki, book, taxonomy): the presentation nodes are set out to represent the knowledge model and the appropriate content – links are bidirectional (unlike the first two use cases) and cross-links through the tree allowed and there is no sequence in terms of end points or logical conclusions. Organization nodes may be used to group or categorize presentation nodes.
4. Algorithm (diagnostic or otherwise): the conditional steps in the algorithm are created as presentation nodes and appropriate content added to each. A user may just use the algorithm to explore or diagnose a problem or a series of questions and counter-scored answers may allow the algorithm to guide the user down the correct path.
5. Simulation (scenario, game, choose your own adventure, game of chance): a decision diagram of the different choices to be given to the learner is created and executed using presentation nodes, appropriate content is loaded in to each and then counters, counter rules, content rules and conditionals are layered in to transform



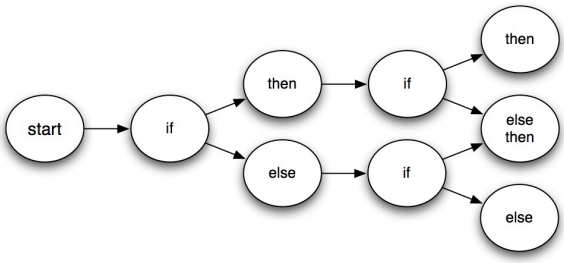
use case 1



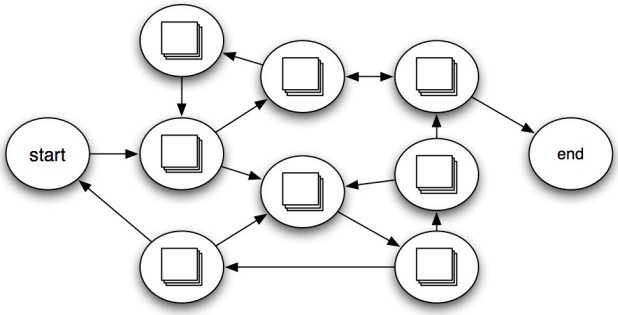
use case 2



use case 3



use case 4



use case 5

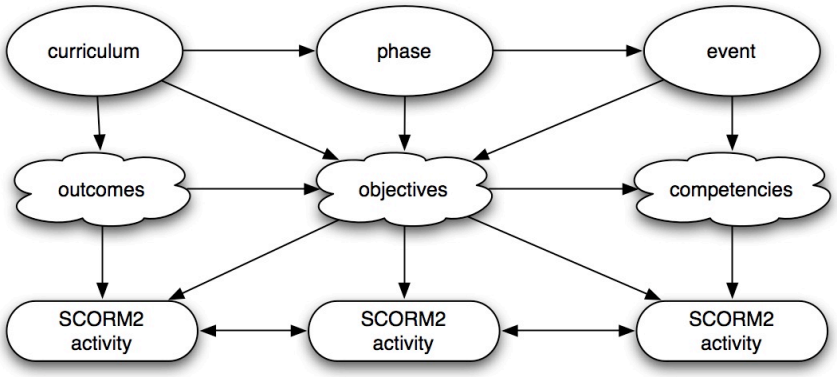
The Big Picture: SCORM2 in an Education Systems Environment

This brings us on to two other related frameworks that enable the use of SCORM2 in a meaningful (rather than simply articulate) way:

- Learner competence: a common framework for expressing what the learner should be able to do as a result of a learning activity. Following a continuum from outcomes (the expected cognitive change for a whole curriculum), through specific objectives to the individual achieved competence of any given learner against those outcomes and objectives.
- Curriculum: a common framework for expressing the sequencing of events and activities across a whole unit or program of study. This should be a model for combining resources (people, space, materials), scheduled activities and outcomes/objectives/competencies.

The following diagram sets out a high-level model for how curriculum (row 1) connects to competence (row 2) and activities expressed as SCORM2 packages (row 3). Note however that these two frameworks are enablers for SCORM2 but are not a part of it. Considering this point of scope and modularity raises important philosophical points around purpose, applicability and utility of SCORM2. The proposed model is an implementation-agnostic model intended to afford

its users the widest range of possible activities that can be expressed using it - they may be simple, complex, good, bad, cruel or surreal but SCORM2 should let you build them. This then allows to separate the discourse of what is good and bad design from the architecture in which it is expressed - ie the journey from the vehicle. Not only does this support good and structured discourse it potentially gives us a kind of abstracted notation (in the same way for math or music) for modeling, evaluating and researching educational activity designs.



Getting to SCORM2

Clearly a lot of this will depend on the overall LETSI process but if this proposal or parts of it go forward to become SCORM2 then the significant amount of technical and conceptual work that has already been done in developing its immediate antecedents can be leveraged here. As a result this model for SCORM2 can be relatively fast tracked into the testing and comments stage. The only significant step to take would be to engage and persuade the wider SCORM community of the viability and applicability of this model.

References

Begg, M., Dewhurst, D. and MacLeod, H. (2005). "Game Informed Learning: Applying computer game processes to Higher Education." Innovate 1(6).

Bloom, B.S. (1984). Taxonomy of educational objectives. Boston, MA, Allyn and Bacon.

Ellaway, R. (2006). "Weaving the ‘e’s Together." Medical Teacher 28(7): pp587-590.

Issenberg, S.B. (2006). "The Scope of Simulation-based Healthcare Education." Simulation in Healthcare: The Journal of the Society for Simulation in Healthcare 1(4): pp203-208.

Miller, G.E. (1990). "The assessment of clinical skills/competence/performance." Academic Medicine 65(supplement): pp S63-7.

Schön, D.A. (1987). Educating the Reflective Practitioner. USA, Jossey-Bass.

About the author

Dr. Ellaway, PhD, is Assistant Dean and Associate Professor for Education Informatics at the Northern Ontario School of Medicine, Sudbury, Canada and Visiting Professor for Education Informatics at St. George's, University of London, UK. She is co-chair of the MedBiquitous Virtual Patient working group, a member of the MedBiquitous Executive and a scholar in the area of educational technology standards and specifications. She writes a regular column for Medical Teacher and is the Chair of the AFMC’s Informatics Resource Group. Her work on developing online learning environments received a Queens Anniversary Prize for Higher and Further Education in 2005. Contact: rachel.ellaway@normed.ca, 705-662-7196